NASA/TM—2002-211802

Variable Delay Testing Using ONE

Joseph Ishac
Glenn Research Center, Cleveland, Ohio

August 2002

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized data bases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at 301–621–0134

- Telephone the NASA Access Help Desk at 301–621–0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076

NASA/TM—2002-211802

Variable Delay Testing Using ONE

Joseph Ishac
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center

August 2002

Acknowledgments

Many thanks to Mark Allman, David Carek, Andrei Gurtov, Will Ivancic, and Vern Paxson for discussion and insight on various issues discussed in this paper. Also, Wes Eddy provided useful information on Solaris' inner workings.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at http://gltrs.grc.nasa.gov

VARIABLE DELAY TESTING USING ONE

Joseph Ishac
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135
E-mail: jishac@grc.nasa.gov

ABSTRACT

This paper investigates the effect of long and changing propagation delays on the performance of TCP file transfers. Tests are performed with machines that emulate communication from a low/medium-earth satellite to Earth by way of a geosynchronous satellite. As a result of these tests, we find that TCP is fairly robust to varying delays given a high enough TCP timer granularity. However, performance degrades noticeably for larger file transfers when a finer timer granularity is used. Such results have also been observed in previous simulations by other researchers, and thus, this work serves as an extension of those results.

1 INTRODUCTION

One objective of the National Aeronautics and Space Administration (NASA) is to be able to communicate with different space assets using standard protocols, such as the Transmission Control Protocol (TCP) [RFC793]. However, communication in a space-based environment introduces several challenges, one of which is the long, variable delays associated with reaching objects orbiting the Earth [RFC3135, HSMK98].[1] Other researchers have performed simulations of such environments, and the work in this paper seeks to extend such simulations using real machines. Specifically, we compare our results to the simulated results found in [AGR00].[2] Also, we validate the software used to emulate portions of the test network. The validation tests are designed to specifically target key properties of the emulator including variable delays.

The environment studied in this paper consists of a satellite that communicates to Earth by means of a second, intermediate satellite. The delay between the two satellites varies over time, since the distance between them changes as they each follow independent trajectories. Also, communication from the intermediate satellite to Earth adds a fixed propagation time, further increasing the overall delay. In addition to the dynamic delay, the effect on performance while adjusting the TCP timer granularity is considered. Adjusting the timer granularity impacts, among other things, the TCP retransmission timer, which is used to gauge the time a sender should wait for positive acknowledgement from the receiver before resending data. Reducing the TCP timer granularity allows the retransmission timer to expire within closer proximity of its intended target, which may increase performance in the event of loss. However, performance is likely to decrease if the retransmission timer fires prematurely.

The arrangement of machines and overall experiment setup is described in section 2 of this document. Section 3 outlines and explains the tests used to validate the setup. Finally, section 4 outlines the results of the experiments, and section 5 details the various conclusions as well as possible future work areas.

2 SETUP

Satellite Tool Kit (STK) version 4.1 was used to replicate several variable delay scenarios found in [AGR00]. Table 1 lists the five scenarios used in the experiments. Also, the access period for each scenario is specified in the table by both the start time and access duration.[3] NASA's Tracking and Data Relay Satellite System (TDRS) is a constellation of geosynchronous (GEO) satellites. One of these satellites, TDRS 5, is the common body among the scenarios. The accompanying body in each scenario is either a low-earth orbit (LEO) or medium-earth orbit (MEO) satellite and is specified as such in the orbit column.

The testbed used in these experiments consists of three machines whose topology is shown in figure 1. OpenBSD version 2.9[4] is used on each endpoint machine. Also, the source machine runs a custom kernel allowing changes to the TCP timer granularity as well as implementing a pair

[1] Note that long delays are not exclusive to space settings, and that the results in this paper apply to any environment with similar characteristics.
[2] In order to provide an equitable comparison to the simulation study found in [AGR00], delay patterns and file sizes from [AGR00] are used.

[3] Access periods are for July 1, 1999.
[4] OpenBSD 2.9 supports SACK [RFC2018], FACK [MM96, MM96sup], timestamps, and window scaling [RFC1323].

of TCP bug fixes.[5] The Ohio Network Emulator (ONE) [ONE] is used on the final machine (represented by the boxed area in figure 1) to emulate the path between the satellites and Earth. ONE emulates the path by passing packets between two different network interface cards (NICs) after applying the appropriate link delays.

Scenario[6]	Bodies	Orbit	Start (hh:mm:ss.ss)	Duration (sec)
6	ISS to TDRS 5	LEO	00:50:53.81	3220.83
9	RADCAL to TDRS 5	LEO	10:44:22.54	13735.82
11	LAGEOS–2 to TDRS 5	MEO	05:44:48.00	29892.49
12	LAGEOS–2 to TDRS 5	MEO	14:32:19.50	12206.99
13	NAVSTAR–01 to TDRS 5	MEO	00:00:00.00	86400.00

Orbital data taken on July 1, 1999

Table 1: Variable Delay Scenarios

For each scenario in table 1, the delay in each direction across the link is computed using a fixed delay of 125 milliseconds (Earth to TDRS). In addition, a variable delay based on the STK orbital information for the given scenario is added to the fixed delay. Each orbital data point is separated by one minute, and thus, the overall delay changes with the same granularity.[7] Other characteristics for the emulated links include an imposed bandwidth restriction of 1.5 Mbs with no intentional corruption or drops. Also, links between ONE and the endpoints are 100 Mbps Ethernet. Finally, the emulated routers carry a queue size of 50 KB each.

Each experiment consists of sending files of various sizes from the source machine (satellite) to the destination machine (Earth). The file sizes used by the sender range from 2,896 bytes to 2,896,000 bytes, with each size separated by an order of magnitude. Using segment sizes of 1500 bytes with 52 byte headers, the resulting transfers consist of approximately 2, 20, 200, and 2000 data packets. Data transfers are done using a modified version of TTCP,[8] which enables a socket option needed in order to maintain synchronization between tests. Also, the advertised window size on both the sender and receiver is set to 240 KB, which is unattainable given the network setup. The larger window size allows the performance to be primarily influenced by the network conditions and not artificially limited by either the sender or receiver [SMM98].

In addition to file size, the sender's TCP timer granularity is varied. Granularities of 500 and 10 milliseconds are studied. The coarse 500 ms timer represents a value found in many TCP implementations, including the version of OpenBSD used. Thus, the goal is to study the effect on performance when using a finer 10 ms timer. Adjusting the TCP timer granularity affects the TCP retransmission timer, and retransmission timeouts (RTO) may become less conservative with finer granularities [AP99]. However, too aggressive of a RTO can reduce TCP's performance, and thus, a minimum RTO of one second is enforced in every experiment [RFC2988].

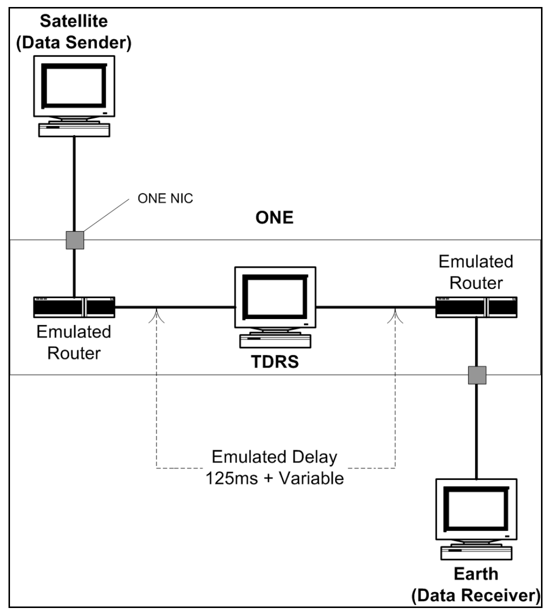

Figure 1: Network Topology

For the duration of each orbital scenario, testing consists of the sender choosing a file size at random and sending it to the receiver. When the transfer completes, the sender waits for a given time based on a Poisson mean of ten seconds and then repeats the process. The test is performed for each orbital scenario and timer granularity. Shorter scenarios are run more frequently in order to reduce sample size differences between scenarios. Finally, tcpdump [Tcpdump] traces packets at both endpoints for each scenario. The sender-side traces are then used to calculate the throughput for each file transfer in order to determine its performance. Also, both traces are used to identify connections that contain spurious RTO's.

5 Bug fixes: PR#2368 and PR#2375, with details at: http://cvs.openbsd.org/cgi-bin/wwwgnats.pl

6 Scenario numbers are non-sequential in order to reflect the equivalent scenario in [AGR00].

7 No linear interpolation is performed between delay values. However, the difference between steps is small (often less than two milliseconds). Thus, the loss in performance to do interpolation justifies not adding such a feature [Kim].

8 "Test TCP (TTCP)". Modified with SO_LINGER support: http://roland.grc.nasa.gov/~jishac/tools/ttcp/ttcp.linger.tar.gz

3 VALIDATION

To assure that ONE was working within acceptance, several validation tests were performed. The majority of these tests involve sending and receiving ICMP echo request and reply packets of various sizes under different configurations of ONE, with each configuration set up to deliberately test a different piece of functionality. A subset of these tests was done in [ACO97]. Overall, tests for proper emulation of queuing, propagation (fixed), and transmission (serialization) delays performed as expected, with results within one percent of the expected values. Therefore, the results of those tests are not included in this document, as they are not critical to the overall topic. Finally, proper emulation of variable propagation delay is also tested, and is covered in the remainder of this section.

Since the ability for ONE to successfully emulate variable propagation delays has not been documented in previous research, a test was devised to compare the expected varying RTT of a given scenario to the RTT achieved using ONE. The resulting test consists of pinging the destination by sending ICMP echo request and reply packets between the two endpoints. The pings are sent for the duration of the scenario with each new echo request initiating a second after the previous one. Figure 2 illustrates the amount of time in seconds that the expected RTT values of scenario nine differs from the values observed in the test. Thus, positive points in the figure represent an observed value which is greater than what is expected. Also, the high density of points located just below one millisecond in the plot can be attributed to the Ethernet delay connecting the endpoints to the ONE emulator, which is not included in the expected value.

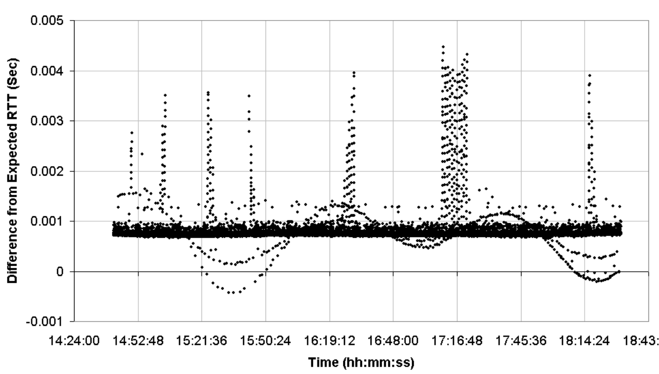

Figure 2: Difference Between Observed and Expected RTT

Figure 2 also shows a few anomalies. First, several spikes in the observed difference appear throughout the test, and are likely the result of issues with the operating system's timer granularity and scheduler as well as the timing structure within ONE. The rational behind this argument is better discussed with the next figure. In any case, the effect appears to be constrained to less than five milliseconds. While this may affect results for links with short delays, the impact is much less severe for longer delays, such as the ones studied in this paper. The second anomaly is the apparent sinusoidal pattern about the median.[9] Since ONE enforces changes to the RTT in fixed intervals, ping requests and replies may traverse two different RTT "steps". While this behavior is correct, the procedure used to calculate the differences never takes such a case into account, using only the later of the two delay steps for comparison. The resulting differences produce the sinusoidal pattern.

Figure 3 shows an overlay of two curves. The first, serrated curve illustrates the observed RTT values. The second, smoother line represents the expected value. The stair-like pattern in the observed curve is a result of the one-minute delay adjustment interval in ONE (as described in section 2). From the figure, we can note that the two curves tightly follow each other, essentially off by the median value (0.73 ms) from figure 2. Therefore, the results support ONE's ability to properly execute variable delays. Also, the frame within figure 3 shows a zoomed view of the observed RTT for the listed time period. The frame gives better insight to the RTT spikes observed in figure 2, as it shows a clear systematic behavior. Again, the cause for such behavior is likely due to timing events within the ONE software, the underlying operating system, or both.

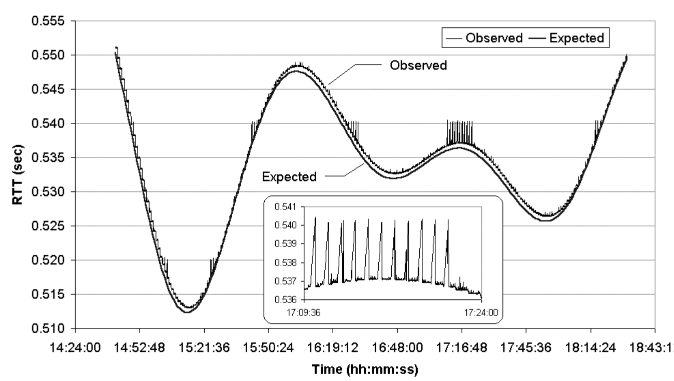

Figure 3: Overlay of Expected and Observed RTT

[9] The pattern only coincidentally resembles the negated first-order derivative of the orbital delay. The actual cause is explained in the text.

4 RESULTS

The results found in this section are based on the test described in section 2. Figure 4 shows the average throughput per scenario for each file size tested and for a timer granularity of 500 milliseconds. Data labels are displayed for each data point, so that the average values are clearer on the log scale. The error bars represent the 5th and 95th percentile for the subset of data that each point represents. Choosing the percentiles is done in order to remove any outliers.[10]

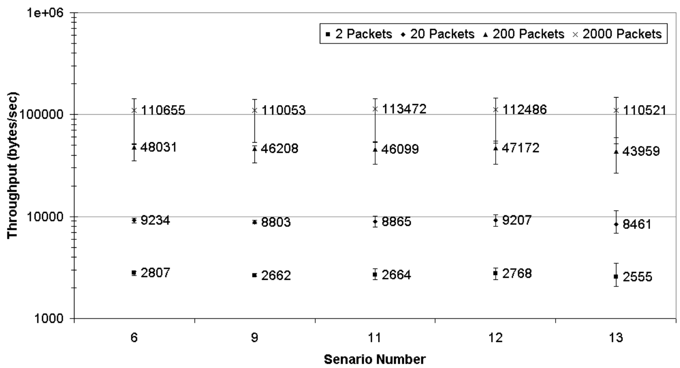

Figure 4: Average Throughput per Scenario and File Size (Granularity = 500 ms)

As expected, throughput increases as the transfer size increases, primarily since the smaller connections are inhibited by slow start. For the largest file size, the available link bandwidth is roughly fully utilized, regardless of the scenario. Also, on average, throughput is not affected by which scenario is used, since many of the orbital patterns fluctuate within similar bounds. Figure 4 further shows that variations in throughput are larger (by roughly 30%) for larger file sizes. This result is in direct contrast to the simulation results found in [AGR00]. The reason for the discrepancy is likely due to the different loss patterns experienced in both experiments. Loss patterns in a simulator often follow a clean and predictable behavior, providing consistent results, which can be easily reproduced. However, the testbed is subject to factors such as scheduling, clocks, and general software flaws which are expected to produce a range of different loss patterns. Some simple tests were successfully run to further back this assumption, however in-depth analysis and verification is left as future work. Variations in throughput are also larger for certain scenarios such as scenarios twelve and thirteen. However, this is expected since those scenarios carry a greater amount of variability in the orbital pattern. Also, while the amount of variation due to orbital patterns can be seen for each file size, the effect is best observed at the smallest file sizes (2 and 20 packets).

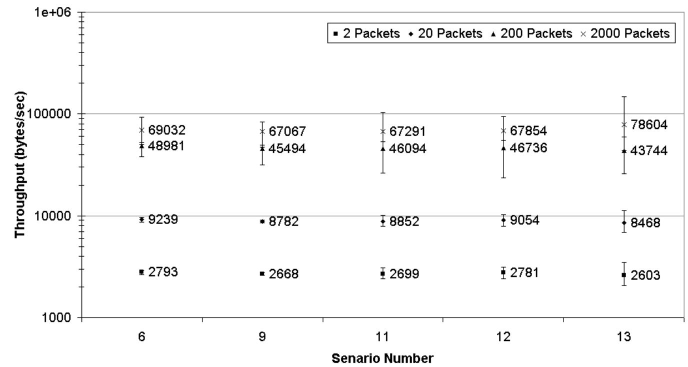

Figure 5: Average Throughput per Scenario and File Size (Granularity = 10 ms)

Figure 5 shows the same information found in figure 4 with exception that the timer granularity is now 10ms as opposed to 500ms. As seen from figures 4 and 5, throughput values remain relatively the same regardless of timer granularity for all file sizes except the largest (2000 packets). Figure 6 better shows the difference in average throughput for 2000 packet transfers at the two granularities. Most of the smaller file transfers do not suffer from loss, and are not large enough to create significant queuing delays. So, the RTT remains below the one second minimum RTO used for all tests. Thus, the effect of granularity changes on the smaller transfers is not significant. However, the larger file sizes are able to build a queue in the emulated routers, causing losses and delay spikes, which slightly exceed one second. As a result, connections may suffer from spurious timeouts, especially when a fine-grained timer is used. To determine the amount of connections that experienced spurious RTOs, a packet pair analysis was performed from full packet traces taken at both the sender and receiver. A RTO is considered spurious if a data segment, which has already arrived at the receiver, is retransmitted (as a result of a RTO[11]) by the sender. Analysis of 2000 packet transfers (for all scenarios) with a 10 ms granularity, showed that 78% of the connections contained a spurious RTO. However, only 11% of the 500 ms connections experienced any spurious RTOs. The 67% increase is a result of the 10 ms timer being more aggressive, firing within closer range of the intended one second mark. The erroneous timeouts result

[10] For example, losing the initial SYN would cause an abnormally low throughput. Such losses did occasionally occur even though the tests were a single flow analysis. The losses are attributed to the ONE bridge occasionally being unable to forward a packet correctly.

[11] Where an RTO is defined as a retransmission that occurs a second or more from the last acknowledgement for outstanding data.

in needless retransmissions and a reduction in the TCP sending rate, which decreases the connection's throughput. In these tests, a decrease in throughput of roughly 37%, averaged over the scenarios is seen. An example of a spurious timeout, and its effect on performance, is discussed in greater detail in appendix A.

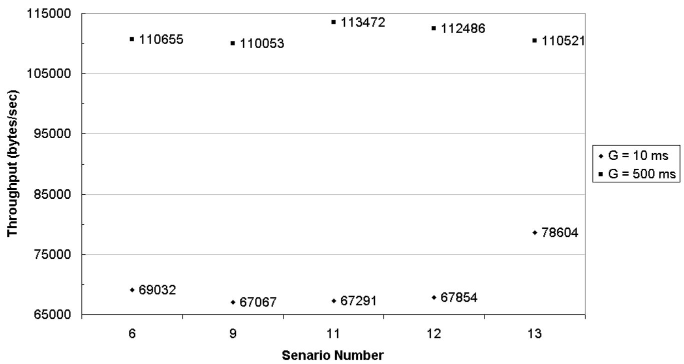

Figure 6: Average Throughput per Scenario for 2000 Packet Transfers

5 CONCLUSIONS AND FUTURE WORK

The experiments in this paper show that TCP is fairly robust to environments with large and varying round trip times, given an ample TCP timer granularity and minimum RTO value. However, throughput degrades significantly for larger file sizes when a finer granularity is used. The decreased throughput is largely due to the presence of spurious RTOs and leads to a decrease in the overall performance. Also, variations in throughput are larger as the file size increases. This result differs from that observed in previous simulations, and is likely due to systematic loss patterns in a simulator, which provide relatively consistent results.

Finally, this work represents only a small fraction of possible tests for varying, long delay environments with possible extensions including:

- Running similar tests with multiple flows.
- Introducing varying signal strength in addition to the variable delay (therefore including losses not caused by congestion).
- Modeling hand-offs between different scenarios.

APPENDIX A

As described in section 4, spurious timeouts reduced the performance of a connection and occurred more frequently when the TCP timer granularity was reduced from 500 to 10 milliseconds. Figure 7 illustrates a transfer that experiences two RTOs for which the first is spurious. The figure shows part of a time sequence graph generated by tcptrace [Tcptrace] for a large 2000 packet file transfer. Reference labels (boxed characters) in the figure are provided for clarity, and the series of events that take place are explained in the list below.

- A large increase in sending rate (not shown) leads to an increase in queuing delays. The increase is due to a large advertised window, a large delay-bandwidth product, and an exponentially increasing congestion window due to slow start.
- The sending rate exceeds the link and buffer capacity and a loss event occurs (Fig. 7–A). Subsequent packets trigger duplicate acknowledgements and trigger a fast retransmission (Fig. 7–B) followed by multiple retransmissions to fill SACK gaps.[12]
- The retransmission timer expires (Fig. 7–C). The expiration time is correct in length—about one second. However, an ACK does arrive several milliseconds after the RTO, which is less than the feasible RTT of the link. Therefore, the ACK must correspond to the fast retransmission, and thus, the RTO is spurious.
- The RTO induces slow start. However, a flood of acknowledgements for the SACK'ed retransmissions causes slow start to increase quickly (Fig. 7–D). The rapidly increased sending rate again overruns the link, and the losses yield another RTO (Fig. 7–E).

The series of events cause a significant loss in performance, and occurs frequently (78%) in these tests when transferring the largest file size using a 10 ms granularity. Furthermore, resolving the problem can become rather complex. For example, one solution would be to reset the RTO timer based on events during loss recovery, such as retransmissions or duplicate acknowledgements. However, care must be taken not to continuously delay the expiration of the RTO.[13] Another difficult solution would be to adjust the way TCP calculates the RTO, making it more robust to such situations. Thus, an adequate solution is difficult, and further discussion and evaluation of such solutions are left as areas for future research.

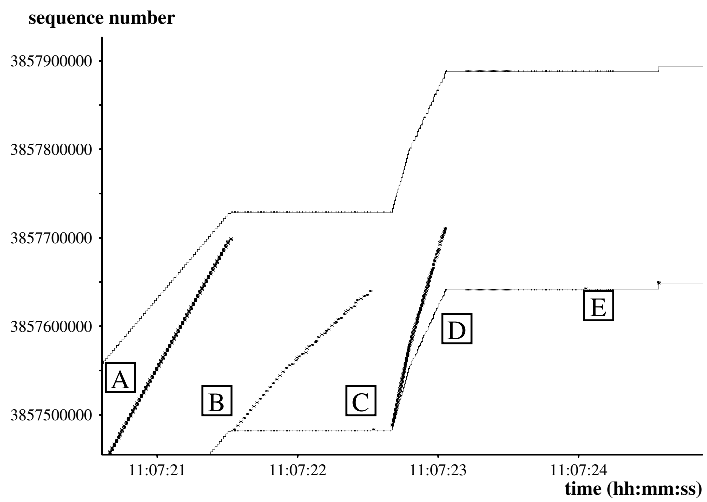

Figure 7: Time Sequence Graph for a double RTO event

[12] SACK holes are filled as new SACK information indicates that enough packets have left the network path.

[13] For example, simply resetting the RTO on duplicate ACKs would lead to problems if the initial fast retransmission were lost. The sender may end up exhausting the receiver's advertised window, before suffering the RTO.

REFERENCES

[ACO97] M. Allman, A. Caldwell, and S. Ostermann, "ONE: The Ohio Network Emulator", Technical Report TR–19972, Ohio University Computer Science, August 1997

[AGR00] M. Allman, J. Griner, and A. Richard, "TCP Behavior in Networks with Dynamic Propagation Delay", IEEE Globecom, November 2000

[AP99] M. Allman and V. Paxson, "On Estimating End-to-End Network Path Properties", ACM SIGCOMM, Sept. 1999

[HSMK98] T. Henderson, E. Sahouria, S. McCanne, and R. Katz, "On Improving the Fairness of TCP Congestion Avoidance", IEEE Globecom, November 1998

[Kim] J. McKim, personal communication, 2001

[MM96] M. Mathis, J. Mahdavi, "Forward Acknowledgment: Refining TCP Congestion Control", ACM SIGCOMM, August 1996

[MM96sup] M. Mathis, J. Mahdavi, "TCP Rate-Halving with Bounding Parameters", http://www.psc.edu/networking/papers/ FACKnotes/current/

[ONE] ONE Project: http://irg.cs.ohiou.edu/one/index.html

[RFC1323] V. Jacobson, R. Braden, D. Borman, "TCP Extentions for High Performance", RFC 1323, May 1992

[RFC2018] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement Options", RFC 2018, October 1996

[RFC2988] V. Paxson and M. Allman, "Computing TCP's Retransmission Timer", RFC 2988, November 2000

[RFC3135] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, June 2001

[RFC793] J. Postel, "Transmission Control Protocol", RFC 793, September 1981

[SMM98] J. Semke, J. Mahdavi, and M. Mathis, "Automatic TCP Buffer Tuning", ACM SIGCOMM, September 1998

[Tcpdump] TCPDUMP Project: http://www.tcpdump.org/

[Tcptrace] tcptrace Project: http://irg.cs.ohiou.edu/software/tcptrace/

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 2002	Technical Memorandum

4. TITLE AND SUBTITLE

Variable Delay Testing Using ONE

5. FUNDING NUMBERS

WU–258–90–00–00

6. AUTHOR(S)

Joseph Ishac

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field
Cleveland, Ohio 44135–3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E–13497

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM—2002-211802

11. SUPPLEMENTARY NOTES

Responsible person, Joseph Ishac, organization code 5610, 216–433–6587.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category: 17 Distribution: Nonstandard

Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 301–621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

This paper investigates the effect of long and changing propagation delays on the performance of TCP file transfers. Tests are performed with machines that emulate communication from a low/medium-earth satellite to Earth by way of a geosynchronous satellite. As a result of these tests, we find that TCP is fairly robust to varying delays given a high enough TCP timer granularity. However, performance degrades noticeably for larger file transfers when a finer timer granularity is used. Such results have also been observed in previous simulations by other researchers, and thus, this work serves as an extension of those results.

14. SUBJECT TERMS

Satellite communication; Delay; Orbits; Data transmission; Transmission efficiency; Networks; Variability; Communication satellites; TDR satellites; Protocol (computers)

15. NUMBER OF PAGES

15

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	